Genetic Algorithm based Layered Detection and Defense of HTTP Botnet

Seena Elizebeth Mathew[1], Abdul Ali[2], Janahanlal Stephen [3]

[1] Computer Science and Engineering Department, Ilahia College of Engineering and Technology, Kerala, India
[1]Email: seena2266@gmail.com
[2, 3] Computer Science and Engineering Department, Ilahia College of Engineering and Technology, Kerala, India
[2]Email: abdulali4u@gmail.com
[3]Email: drlalps@gmail.com

***Abstract*— A System state in HTTP botnet uses HTTP protocol for the creation of chain of Botnets thereby compromising other systems. By using HTTP protocol and port number 80, attacks can not only be hidden but also pass through the firewall without being detected. The DPR based detection leads to better analysis of botnet attacks [3]. However, it provides only probabilistic detection of the attacker and also time consuming and error prone. This paper proposes a Genetic algorithm based layered approach for detecting as well as preventing botnet attacks. The paper reviews p2p firewall implementation which forms the basis of filtering. Performance evaluation is done based on precision, F-value and probability. Layered approach reduces the computation and overall time requirement [7]. Genetic algorithm promises a low false positive rate.**

***Index Terms*— HttpBotnet, DPR, Firewall, Layered approach, Genetic Algorithm.**

I. Introduction

Network security is the grand edifice of internet security and it involves policies and principles to enforce security in the network. Nowadays most of the security attacks are conducted by means of bots. Bot is a system that is already compromised by an attacker. Botnet refers to a collection of systems (or application programs) communicating with other systems to initiate attacks controlled by the attacker [1]. Botnet stems from the words rebot and networks.

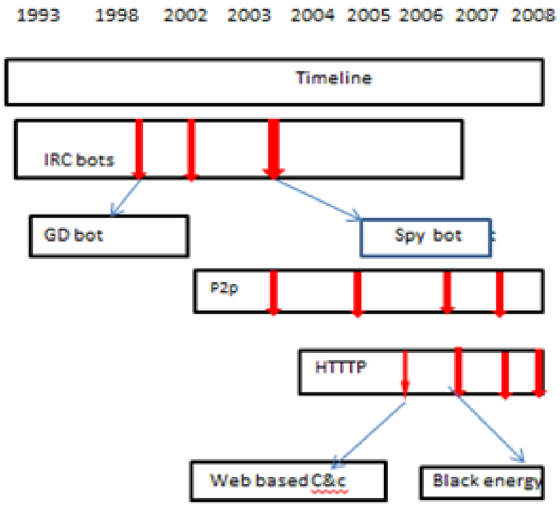

Figure 1. History of malicious bots [1]

In the earlier days bots were the authorized tools used for controlling IRC channel. Soon they changed to malicious bots. They are becoming a major tool for cybercrime, because they can be designed to disrupt targeted computer systems in different ways very effectively, and because a malicious user, without possessing strong technical skills, can initiate these disruptive effects in cyberspace by simply renting botnet services from a cybercriminal [4].A representation of history of malicious botnets is shown in Ref. [3]. However in the figure 1 the history of evolution of botnet has been highlighted. IRC bots appeared on the internet by 90's and it is developed to P2P bots and recently it is evolved to HTTP bots [3].

Background knowledge on aspect of DNS protocol relevant to botnet detection is essential. As Don Marti once observed, "DNS is a consensus reality[16]. The type of the server (authority server) depends a lot in mapping between domain name and IP address. Different host will get different result for the same domain because of the variations in caching. The DNS infrastructure, and resolvers like BIND[17], trying to solve this issue; however, bot masters have crafted their networks to leverage this potential. The basic principles of botnet scenario are explained in chapter 10[18].

The proposed system uses genetic algorithm based layered approach for detecting http botnet. The system makes use of four autonomous and self-sufficient layers(probe layer, DDoS layer, U2R layer, R2L layer). Each layer is responsible for detecting different attacks and makes use of threshold value for the attack detection. These threshold values are generated by genetic algorithm. Once a layer detects an attack further request from that IP is automatically filtered by firewall filtering unit.Genetic algorithms are searching techniques based on natural selection model (preservation of favorable features and rejection of others) to find the optimized solution. Genetic algorithm begins with an pool of possible solution and evolves them to form most suitable solution through a series of steps.

There are different classifications on Genetic algorithm which can be applied on botnet for diverse reasons. They are parallel genetic algorithm, distributed genetic algorithm, hybrid genetic algorithm, and simple genetic algorithm.

Parallel genetic algorithm or PGA can be implemented by executing SGA (simple genetic algorithm) in parallel. Hybrid genetic algorithm in-cooperates genetic search and other local search techniques to enhance the performance and accuracy. The basic concept of adaptive genetic algorithm is to vary mutation probability (P_m) and crossover probability (P_c) according to the individual fitness. More optimized results can be obtained by dynamically adjusting P_m and P_c .

The proposed system uses simple genetic algorithm since it involves only copy operation and exchange of certain parental features. The algorithm promises more optimized results with less computational complexity.

II. Related Work

A. Automated Layered Approach for Layered Detection

Detection is carried out in a layered fashion. A manual threshold is set initially for every layer. If the number of packets to the layers is greater than a particular manual threshold value then corresponding attack is reported and database is updated accordingly. Once detected, further request from that IP address is automatically filtered by the filtering unit [1]. Detection and performance are depending on the manual threshold value. Accuracy of manual threshold value is significant.

B. Genetic Algorithm Based IDS

Genetic algorithm for intrusion detection system consists of two phases [2]:

1) Pre-processing and feature extraction phase

Pre-processing and feature extraction phase selects two training and testing sets randomly from the given set of population or from the existing database, convert symbolic features into numeric ones, normalise the population, extract the population with better fitness.

2) Training and Testing Phase

Training and testing phase involves initialising the population. Selecting new population randomly from initial population, evaluate new population, and apply genetic operators(mutation and crossover) to new population pool until population with required fitness is reached [2].

C. Botnet Detection Based on DPR

A measurement may be said to be repeatable when the variation in the measurement is smaller than some agreed limit [13]. This repeatability is usable for analysis of HTTP clients and servers. Degree of periodic repeatability is used to describe relationship between them and use degree of freedom and standard deviation to calculate repeatability standard deviation [14, 15]. If DPR of a connection is less, it indicates that the repeatability is higher and that may be attacker’s connection[3]. DPR based detection provides only a probabilistic detection of the attacker and it is time consuming. DDoS attacks only can be detected through this method.

D. DNS Traffic Based Botnet Detection

Choi [5] proposed the botnet detection by monitoring group activities in DNS traffic. He analysed the different features of botnet DNS and legitimate DNS. He said that the botnet can evade our algorithms when the botnet uses DNS only at initializing and never use it again (e.g., some HTTP botnets) [5].

E. Layered Intrusion Detection

LBIDS [7] provides a layered approach for intrusion detection. Security attributes such as confidentiality, integrity and availability are checked at each layer. Detection is done sequentially.

For detecting breach of confidentiality, integrity and availability, each layer needs to evaluate only specific features of the file essential for the detection.

F. Genetic Algorithm and Classification

Genetic algorithms are simple but provide most efficient results for the problem domain based on natural selection These search algorithms follows a step-by –step paradigm to find a pool with better fitness.

Moreover Genetic algorithms are based on genetic models and hence computationally less complex. Genetic algorithms are gifted with many features. These features enable to form various classes of approach for genetic method. Simple genetic algorithm, parallel genetic algorithm (PGA), distributed genetic algorithm (DGA), adaptive genetic algorithm, messy genetic algorithm and so on [6]. Our proposed system makes use of simple genetic algorithm.

III. Problem Domain

A. HTTP Botnet

Computer security means security of computer based equipment, information (information security), service it provides and the security of associated networks (network security). Computer security is critical in all computer based applications.

Network security refers to the principles and mechanisms undertaken to ensure confidentiality integrity and availability. Security components include Antivirus software packages, secure network infrastructure, Virtual private networks, Identity services, Encryption and Security management. Security can be breached by hackers and other type of prominent threats. Nowadays bots play a vital role in network attacks.

When a computer is compromised by penetrating malware software, then its control goes to bot master (the third party who has conducted the attack). The bot master then can able to control the activities of bots through IRC channel or HTTP. Botnet attack method is given in the Fig. 2. HTTP botnet uses HTTP protocol. HTTP bots developed from P2P bots which in turn evolved from IRC bots. Bots exploit DNS servers for evasion. Features of bot’s DNS traffic are different from normal one and thus it enhances the detection.

The proposed system makes use of a layered approach using genetic algorithm concepts for HTTP botnet detection. This Anomaly detection employs four autonomous and self-sufficient layers (Probe layer, DDos layer, U2R layer and R2L)

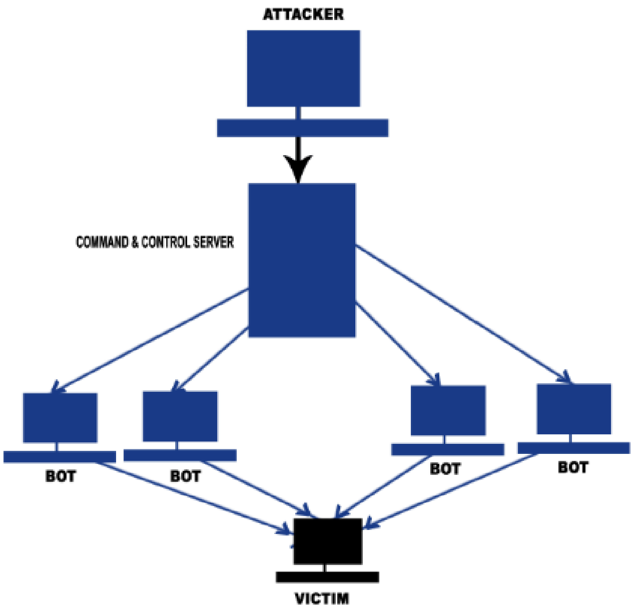

Figure 2. Botnet Attack

to detect different attacks conducted by bots. The overall performance of the system is based on threshold values which are generated by genetic algorithm.

B. Simple Genetic Algorithm

Simple genetic algorithms involve only copy operation and exchanging certain features of parents. It promises more optimised results with less computational complexity. The operations involved in SGA are:

- Reproduction
- Mutation
- Crossover

Reproduction is nothing but copying certain individuals from the initial population based on fitness function. That is individuals with more adaptability are chosen to form new set of pool. SGA adopt different selection[8,9] techniques. After generating selection pool, apply genetic operators. At first crossover is applied by selecting two parents randomly from the selection pool, breeding them to form new offspring. Then mutation is applied by selecting individuals from mutation pool and mutating certain features. Reasons for choosing SGA in the proposed system are:

- SGA's are simple and easy to implement
- Higher adaptability
- Less computational complexity

C. Motivation

The motivation is to prevent following attacks.

1) Port Scanning

Port scanning attack refers to the scanning of ports to identify available services and the active ports in the systems.

2) Distributed Denial of Service Attack

Distributed denial of service attack aims at flooding the system with illegitimate request more than it can service.

3) HTTP Error Attack

HTTP error attacks can be conducted either by sending an HTTP request having only header part or by sending packet with undefined function.

4) Urgent Pointer Attack

Urgent pointer attacks can be initiated by sending packets with urgent pointer enabled. These packets have no data part.

5)Reset Attacks

Reset attacks refer to sending packets with reset flag enabled. The main purpose of these attacks is to make the server busy.

Proposed system makes use of a layered approach for HTTP botnet detection. It also in cooperates genetic algorithm for finding threshold values for each layer.

IV. Proposed System

A. Problem Statement

Bots sneak into a system by the malware software, perform automated tasks in the compromised systems and pass the control to the bot master who has initiated the attack. Botnet uses HTTP protocol for conducting attack. HTTP protocol and port number 80 evade the attack attempt and pass through firewall without detection [3]. If there is no prevention mechanism it leads to more serious and malicious attacks.

In Ref. [3] the paper deals with distributed denial of service where as in this paper various attacks such as port scanning, DDoS, session hacking, urgent pointer, reset flag attacks etc are addressed to be prevented.

B. Problem Formulation

Differentiate between legitimate packet and bot packet and block further request from the bot.

```
    If (the number of packets < Genetic Threshold AND IP
not in Custom Black list or Black list)
then ACCEPT
    else if (IP not in Black list or Custom Black list )
  {
        ADD to Gray list;
  }
  else if ( in Gray list)
  {
        ADD to Black list;

   }
   else if (in Black list)

   {
        Block
   }
```

1) Solution Methodology

The proposed system makes use of genetic algorithm based layered approach for detecting as well as defending all the attacks narrated above from the HTTP bot.

The system incorporates four layers for detecting four types of attacks and these layers are independent. Hence communication overhead between the layers is very less.

Layered approach reduces the time requirement and complexity [2].

In the case of HTTP botnet, the method that the bots adopt for conducting attacks changes from time to time. That's bots introduce some sort of dynamicity in their operation. Thus, a solution method, robust to the dynamic changes in the environment is highly required.

Therefore the proposed system makes use of genetic algorithm which is an efficient evolutionary technique. Evolutionary algorithms mimic natural selection for finding optimal solution. Principle of natural selection is drawn from adaptation.

Genetic algorithms [11] are searching techniques based on natural selection model (preservation of favorable features and rejection of others) to find the optimized solution. Flow diagram of genetic algorithm is given in Fig. 3.

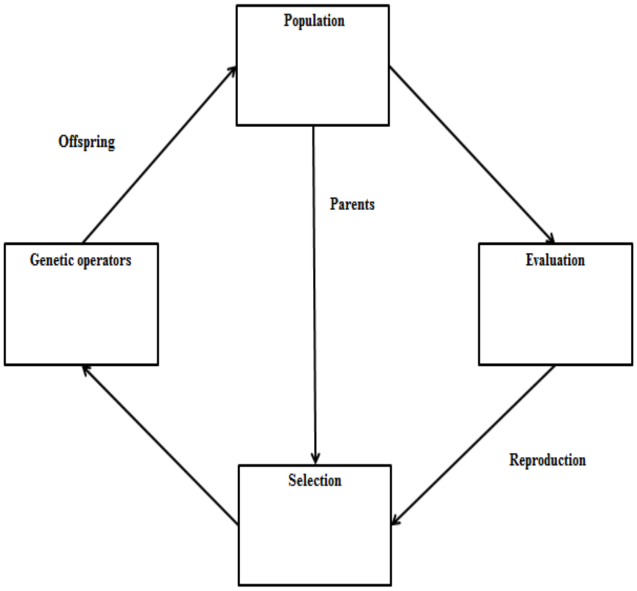

Figure 3. Flow diagram of Genetic Algorithm

Genetic algorithm starts with an initial population. Then form a selection pool. After that it selects parents from the population randomly and breeding them to form new offspring. Genetic operators such as crossover and mutation are applied to the selected individuals.

New offspring promises better characters desirable to form more optimised solution. Some degree of randomness is introduced to make it stochastic. Then fitness is calculated to evaluate adaptability of each individual. Individuals with better fitness are included in the next pool.

The advantages of Genetic algorithms in the study are [6]:

- Simplicity in implementation of HTTP botnet defence
- Robust to dynamic environment since genetic threshold can value varies in accordance with the current scenario.
- Low False positive rate because number of false detection is less.
- Optimised solution is due to fact that genetic algorithm always produce the best result.
- Considers a pool of solutions rather than a single solution in the case of HTTP botnet detection.
- Allows randomness(Stochastic, random selection of individuals from the given population) because here genetic algorithm considers the input randomly from user history Less computational complexity due to the fact that, the proposed system makes use of SGA and mutation and crossover probability doesn't change dynamically.

The reason why AGA is not considered is due to the fact that AGAS are natural selection heuristics, where population size, mutation and crossover probability change occurs during successive iterations of genetic algorithms. The concept of adaptive genetic algorithm is to vary mutation probability (P_m) and crossover probability (P_c)according to the individual fitness.

The proposed system uses simple genetic algorithm. simple genetic algorithm is simple in a way that it doesn't vary the values of pm and pc dynamically. Thus it reduces the computational complexity

2) Justification for using Genetic Algorithm

Schema is a template which includes a subset of threshold values for each layer (DDoS). The order of a schema $o(H)$ is defined as the number of populations in each template. The length of schema, $\delta(H)$ is defined as the difference between highest and lowest threshold values in a template. By using Holland's [10] equation, it can be prove that average fitness of schema increases exponentially in successive iteration.

$$E[m(H,t+1)] \geq \frac{m(H,t)f(H)}{a_t}[1-P] \quad (1)$$

Where $m(H,t)$ is the number of positive detection due to the threshold in the schema at the time t, $m(H,t+1)$ is the number of positive detection at the time $t+1$. The probability of distribution P is the probability that crossover or mutation destroy the schema H. P Can be expressed as:

$$P = \frac{\delta(H)}{l-1}P_c + o(H)P_m \quad (2)$$

Where $o(H)$ is the order of the schema, l is the number of rows considered. P_m is the probability of mutation and P_c is the probability of crossover.

C. Schematic Model

A block diagrammatic model of the proposed genetic algorithm based layered detection system is shown in Fig. 4.

1) Layered HTTP Botnet Detection

The algorithm used for Layered HTTP Botnet detection is given in Fig. 5.

2) Packet Capturing Module

Packet capturing module intercepts or logs the traffic passing through the ports. Captured packets are analysed

© 2014 ACEEE
DOI: 01.IJNS.5.1.4

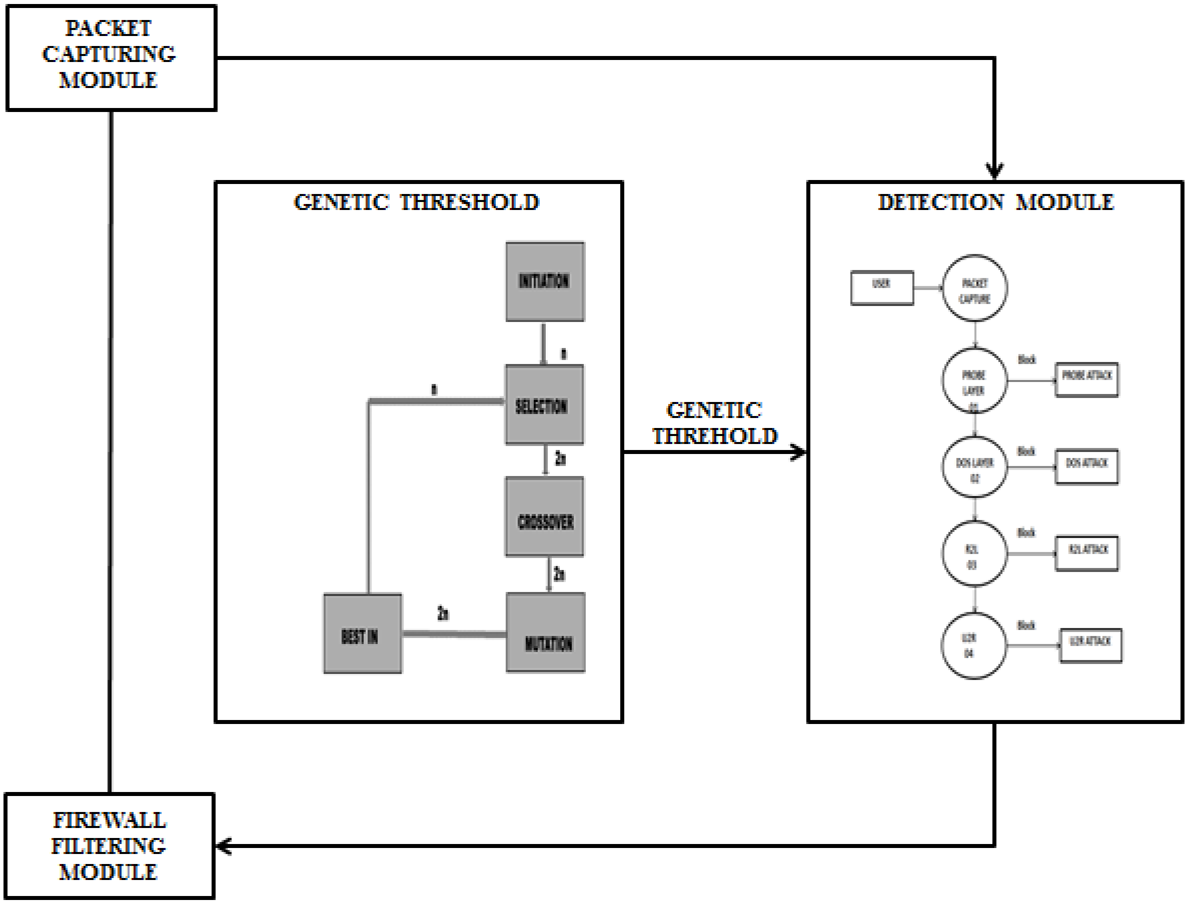

Figure 4. Genetic Algorithm Based Layered System

Step 1: Capture the raw packets from the network
Step2: Calculate Genetic threshold values for each layer
Step 3: Perform Layered Detection based on Genetic Threshold values.
Step 4 : Update the lists based on the Detection.

Figure 5. Layered HTTP botnet Detection Steps

to inspect the raw data. The proposed system employs filtered capturing where packets are captured based on IP address aided with lists it maintain.

3) Genetic Algrithm for HTTP Botnet Detection

The flow chart of Genetic Algorithm for botnet detection is as in Fig. 6.

It includes the following steps.

a) Initialization

An initial population of n set of manual threshold is generated randomly for all the attacks for allowing all possible solutions.

b) Selection

During each successive selection phase the existing population (n rows) is get copied from initial pool. Two parent rows from existing population is selected randomly. Selection is carried out through a rank fitness function. The row with highest rank is set as the n+1 th row. Here input will be n rows and output is of 2n rows.

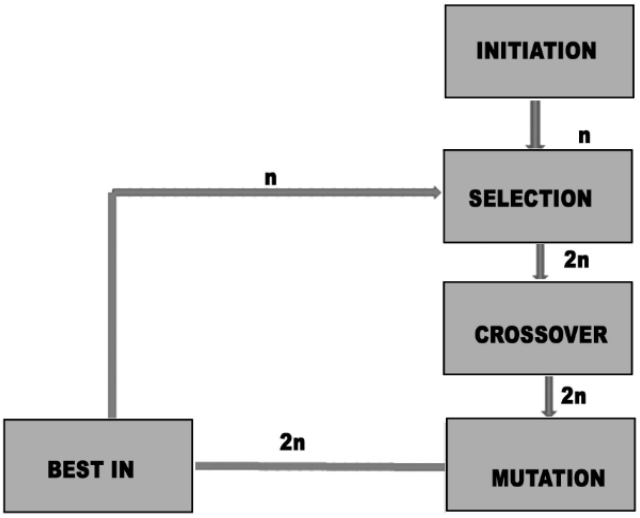

Figure 6. Genetic algorithm Flowchart

c) Cross Over

Genetic operators called crossover and mutation leads to new generation population with better fitness. Copy the first n rows from selection population. For each new population, pair of parent is selected to breed from selection pool and thus new child shares the characters of both the parents. Crossover continues until the population with 2n size has reached. Input and output are of 2n rows respectively.

d) Mutation

Genetic operator called mutation preserve diversity by altering the solutions over generations [16]. For successive iteration, copy the first n rows from previous solution and generate the next population by changing the feature of the randomly selected row. Here input and output will be of 2n rows.

e) Best N Selection

It is worth mutation operators ultimately result in a solution with finer characteristics. Best solution with size n is generated is by calculating the ranks of 2n rows from mutation pool and then sort. Selection of highest rank rows results in next generation solution (n size).

$$Rank\ /Weight\ = \frac{(Success - Failure)}{TotalNumber} \qquad (3)$$

This four layer schemata is continues until fixed number of generations is reached.

Here we define schema H is a template with threshold values for 4 layers. From Holland Schema theorem[14], Expected number of positive detection with schema H at time t +1, in the absence of crossover or mutation, $E[m,(H,t+1)]$ is

$$E[m(H,t+1)] = \sum_{x \in H} \frac{f(x)}{a_t} \qquad (4)$$

$$\sum \frac{f(x)}{a_t} \text{ P\% } \left(\left(\frac{u(H,t)}{a_t} \right) * m(H,t) \right) \qquad (5)$$

Where $f(x)$ is the fitness of x (threshold value), a_t is the average fitness of a function at time t. $u(H,t)$ is the normalised fitness of H at time t. While considering crossover in the algorithm, crossover surviving probability of H, is:

$$S_c(H) \geq (1 - P_c) \frac{\delta(H)}{l-1} \qquad (6)$$

$\delta(H)$ is defined as the difference between highest and lowest threshold values in a template. l is the number of rows considered and P_c is the probability of crossover. While considering mutation in the algorithm, mutation surviving probability of H, is:

$$S_m(H) \geq (1 - P_m) o(H) \qquad (7)$$

Where $o(H)$ is the order of the schema, l is the number of rows considered. P_m is the probability of mutation. On considering mutation and crossover together

$$E[m(H,t+1)] \geq ((\frac{u(H,t)}{a_t})*m(H,t))*(((1-P_c)\frac{\delta(H)}{l-1})*((1-P_m)o(H) \qquad (8)$$

4) Detection Module

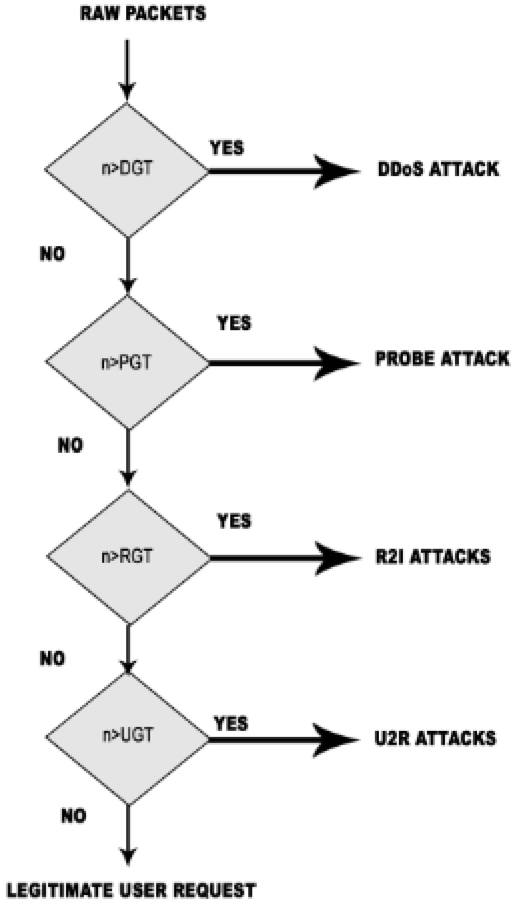

Figure 7. Layered Detection

DGT:-DDoS layer genetic Threshold value
PGT:- Probe layer genetic Threshold value
RGT:- Root to user layer genetic Threshold value
UGT:- User to Root layer genetic Threshold value

Detection is the inclination to presume the purposeful intervention of packets from bots. It involves four layers of defence that resist rapid attempts by the attacker but yield security rather than exhaust by time consuming tactics. Detection module use the four layers (DDoS, Probe, R2L, U2R)which are meant for detecting DDoS attacks, port scanning attacks, reset flag attacks, urgent pointer attacks R2L and U2R attacks. If the number of packets is greater than the corresponding genetic threshold values, these layers report the corresponding attack and database is updated accordingly. The flow chart of layered detection is given in Fig. 7.

a) DDoS Layer

Detection module aims at detecting attacks especially DDoS attacks. In DoS attacks, the targeted system is flooded

DOI: 01.IJNS.5.1.4

with large number of request beyond it can accommodate. Thus it slows down the system's performance. Therefore traffic level features such as source address, destination address, mac address, IP address traffic rate etc. and packet level features such as contents of packets, errors in the packets are considered. System maintains a genetic threshold value. If the number of packets is greater than the threshold, this layer reports DDoS attacks and database is updated.

b) Probe Layer

Different probe attacks this paper considers are request from the client with reset/set request by setting the reset flag, http request with no data, Session hacking, port scanning etc.

Session hacking refers to the theft of session key that is used to authenticate a session. So that, attacker can access to the information exchanged in the session without authorisation. In session fixation attacker fixes the session key but the parties involved in the session are unaware of it. Next one is Session side jacking here the attacker reads the network traffic by packet sniffing. Once the attacker gets the session key he can directly access the data without intervention.

Port scanning is process of scanning the ports. Since data to and from the system goes through the ports administrators uses port scanning for managing the network. Attackers use port scanning for analysing the system traffic. Port scanning is mainly used for identifying the active ports and exploiting the service provided by that port .It is done by sending infinite number of request to a large set of ports. Port sweep is yet another variation of port scanning. In reset attacks, the attacker sends the packets with reset flag enabled without having a data part. It is done to slow down the performance of the server system.

c) R2L Layer

The R2L attacks are one of the most difficult to detect as they involve the network level and the host level features. Therefore both the network level features such as the “duration of connection” and “service requested” and the host level features such as the “number of failed login attempts” are considered for detecting R2L attacks.

Attackers do not have an account on the victim machine, so they trying to gain access.eg. Password guessing.

d) U2R Layer

The U2R attacks involve the semantic details that are very difficult to capture at an early stage. Such attacks are often content based and target an application. Hence, for U2R attacks, we selected features such as “number of file creations” and “number of shell prompts invoked,” while ignores features such as “protocol” and “source bytes”. An attacker has local access to the victim machine and tries to gain super user privileges.

5) Firewall Filtering Module

Firewall performs filtering operation based on black list, white list etc. [3]. The following DFD in Fig. 8 indicates the operation of firewall filtering module.

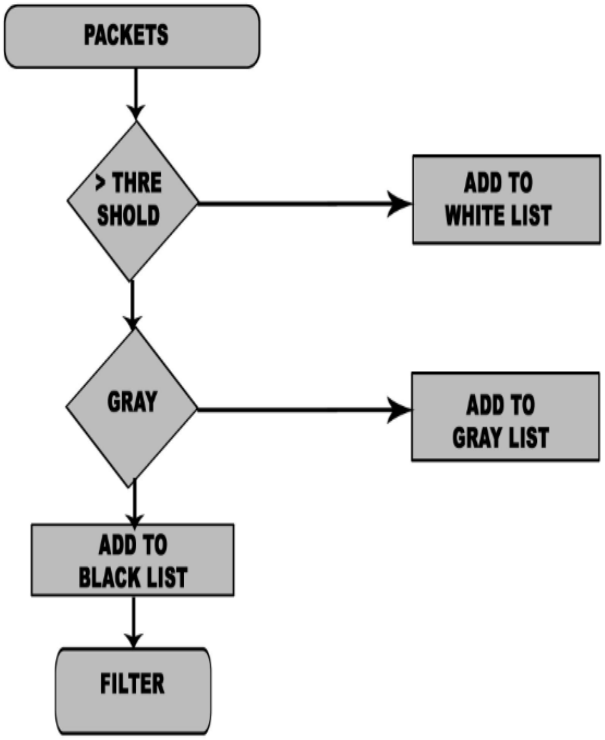

Figure 8. Firewall filtering

a) Custom White List

Users can initially set their white list with the IP address of the other systems they believe to be reliable or authenticated. For this user keeps a database table consisting of IP address and corresponding status. User can add IP address to the custom white list so that connection from such IP is always accepted without intervention.

b) Custom Black List

Users can also initialise custom black list with the illegitimate IP address. Once an IP is included in the custom black list further request from the IP is automatically rejected by the firewall filtering module.

c) Gray List

When number of packets from a particular IP is above a genetic threshold value for the first time, that IP is added to the gray list and starts monitoring its activities.

d) White List

White list is updated after the analysis of the packets. Packets are checked under differ layers, if packet is legitimate packet then source address in the packet is added in white list. For this also a database table is used with entries for IP address and status.

e) Black List

Black list includes the IP address of the system which has been proved to be an attacker. Connection request from the IP's in the black list is filtered without analysis. Thus time can be saved.

DOI: 01.IJNS.5.1.4

V. Simulation

A. winPcap

WinPcap is a software that allows to capture the raw packets, transmit the raw packets, and also to filter the packets based on user specified rules. It has got the privilege for low-level access . Low-level access is achieved by means of a driver which extends the operating system so as to attain that privileges. So that it can make a semantic analysis of receiving packets. It can also note the time at which packets arrived. WinPcap software is an industry tool for network analysis.

B. JPcap

JPcap is a open source software. It has features which enable capture of any transmitting packets for java based application. It runs with the aid of WinPcap in windows environment. JPcap software like WinPcap allows to capture the raw packets, transmit the raw packets, and also to filter the packets based on user specified rules. It can also identify the internet protocol address of receiving packets, flags (reset, urgent) enabled. It can also determine the arrival time of the packets.

C. Colasoft Packet Builder

A packet generator or packet builder is a type of Colasoft packet builder is software that generate the packets according user requirement and build custom packets. It includes an insert option, this allows to inset a TCP/UDP/ARP packet in to the network. It also provides option for sending individual packets or a set of packets. The table I depicts various tools used for diverse purposes.

Table I. Tools-Process Used

Process	Tools Used	Remarks
Packet Capturing	WinPcap, JPcap(Java Net beans)	Capture the packets with low-level features
Database storage	Wamp Server 2.1	Store receiving packets with its low-level features, genetic Threshold values,
Packet Building	Colasoft Packet Builder s/w	Build the packets with user specified features

VI. Data Model

An input-output model (both system and detection) in tabular form is provided in Table II and Table III where the input data is related to the output data through the processes for easier understanding.

VII. Results and Analysis

When the packet enters to the network, firewall first checks whether the sender's IP is included in Black list or Custom black list, if not the packet is captured by packet capturing module with the help of a tool called JpCap. Otherwise firewall blocks the packet. After capturing, these packets are send to the detection module, where detection is carried out in a layered fashion.

A. System Input-Output Model

Table II. System Input-Output Model

Input	Process	Output
Packet History	Initialization	Initialization pool
Initialization pool	Selection	Selection pool
Selection pool	Mutation	Mutation pool
Mutation pool	Cross Over	Cross Over pool
Cross Over pool	Iteration check	Genetic Threshold Values
Raw Packets	Layered Detection	Bot/Legitimate user
Bot system ip	Firewall Filtering	Blocking the Attacker

B. Detection Input-Output Model

Table III. Detection Input-Output Model

Input	Process	Output
Raw Packets	Probe Layer Check	Probe attack Detection
Raw Packets	DoS Layer Check	DoS attack Detection
Raw Packets	R2L Layer Check	R2L attack Detection
Raw Packets	U2R Layer Check	U2R attack Detection

Detection module includes mainly four layers such as DDoS layer, Probe layer, U2R layer, R2L layer. Each layer is meant for detecting different types of attacks. For detecting these attacks, each layer maintains a genetic threshold value, which is generated by Genetic Threshold module. Genetic threshold module makes use of a genetic algorithm for calculating genetic threshold values. In the case of HTTP botnet, the way that the bots adopt for conducting attacks changes from time to time. That means bots introduce some sort of dynamicity in their operation. Thus, a solution method, robust to the dynamic changes in the environment is highly required. Hence Algorithm in the proposed system takes input values randomly from the user history. As a result genetic threshold values vary with time even for the same number of iterations.

Let n be the number of iterations of genetic algorithm. For n=4, a set of genetic threshold values are shown over figs.

Fig. 9, Fig. 10.

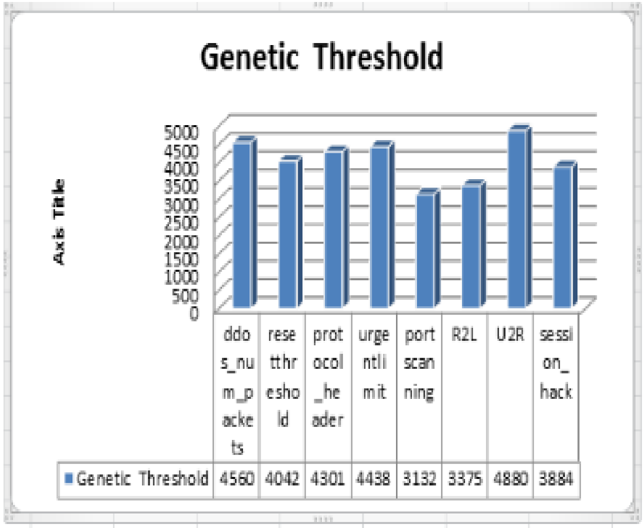

Figure 9. A set of Genetic Threshold Values for n=4

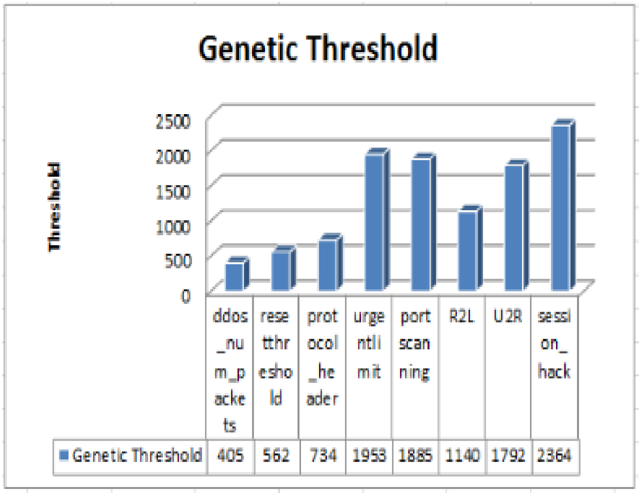

Figure 10. Another set of Genetic threshold values for n=4

The above two figures show the graphs that are plotted at t=0,t=1 for n=4 and we notice the change of threshold for a particular type of attack for a particular number of iteration n=4 is due to the fact that the algorithm takes different input values at ti and also because of mutation and crossover operators applied to the population. The input value is nothing but the user history table.

These changes are expected because of the dynamicity introduced by the bots in their attack pattern. Dynamism is due to the fact that the pattern of the attack is different since the number of packets send on each time may be different.

The experiment is repeated for various other values of n=8, 5, 12 etc. and similar results observed.

If we increase the number of iterations, the false positive rate is improved due to the properties of genetic algorithm. The advantage of such an improved false positive rate is that no legitimate user is tagged as an attacker.

A. Optimization of Threshold Values

During successive iteration, weight (fitness) of the resulting solution pool is increasing exponentially as shown in the Fig. 11. The inclination in the weight is due to fact that the genetic algorithm will always results in more optimized solution population.

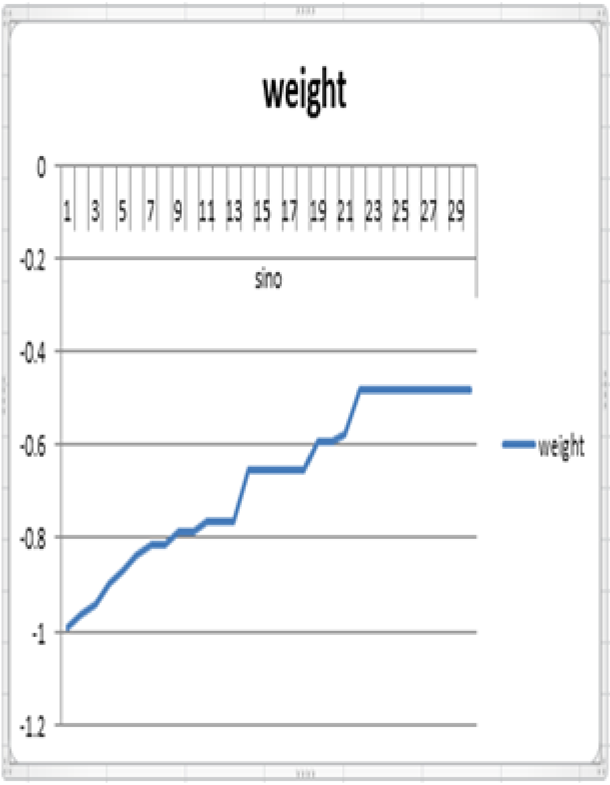

Figure 11. Optimization of Fitness Function

B. Performance of Layered Detection

After generating the threshold values, module sends these values to the corresponding layers of detection module to enhance the detection. First layer(DDoS) compares the number of packets with DDoS genetic threshold, if it is greater, DDoS layer reports DDoS attack and includes senders IP in the black list and the detection stops. Otherwise it send the packet to the next layer that is the probe layer. The same procedure is repeated for probe layer,U2R layer and R2L layer. If no attacks can be detected until R2Llayer, then it indicates that the sender is a legitimate user so that the system can process that packet.

The following Table IV indicates the performance of layered approach. It is calculated using the comparative study of Precision, Recall and F-Value on Probe, DoS, R2L, U2R layer. Precision can be defined as the closeness of the observed value and the experimental value. Recall refers to the measure of retrieved relevant documents. F-Value is the Harmonic-Mean of Precision and Recall. Number of correct detection by total number experiments gives probability of attack detection

$$F - value = \frac{1}{(\frac{1}{2}(\frac{1}{\Pr ecision} + \frac{1}{\operatorname{Re} call}))} \qquad (9)$$

DOI: 01.IJNS.5.1.4

TABLE IV. PERFORMANCE BASED ON PRECISION, RECALL, F-VALUE

Attack Group	Precision	Recall	F-Value	Prob. Of attack Detection
Probe	1	0.8	0.888889	0.8
DoS	0.8	0.8	0.800002	0.16
R2L	0.8	0.6	0.72727272	0.032
U2R	0.571428 5	0.5714	0.5714285	0.0064

Graphical representation of the performance is shown in Fig. 12.

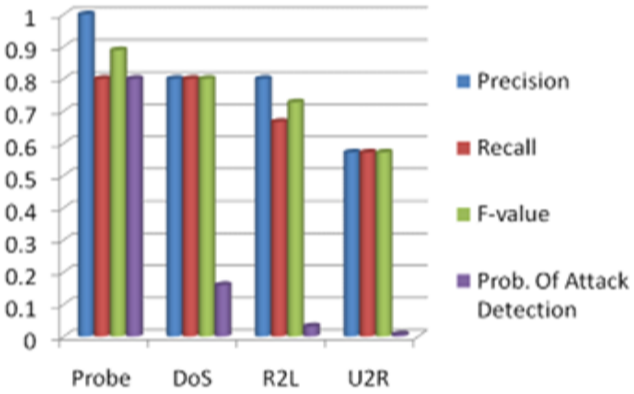

Figure 12. Attack Detection by Various Measures [1]

Once a layer detects one IP as an attacker, that IP is included in the Black list. So that further request from that IP is automatically filtered by the firewall filtering module. Thus it secures the network and system from bots attack.

C. *Tools used to Generate the Graphs*

In warmserver there is an option for export the data to excel. This feature has been used to generate the graph. The coordinate system has been generated by an option available in warmserver. The necessary coordinate values have to be provided.

D. *Complexity Analysis*

The Running time of the proposed system depends on time complexity of genetic algorithm [12] and layered detection.

1) Analysis of Genetic Algorithm

The proposed system uses active termination. So the system terminates after n iterations. Let the number of iterations of the algorithm be n times.

Initialisation occurs only for one time regardless of the number of iterations. Thus the total running time of Initiation is $o(1)$.

Selection only selects a sub population of size n from the given population. Thus the running time of selection for one iteration will be of the order $o(1)$, hence the total running time of selection for n iterations is: $o(n)$.

Next process is crossover. It selects two parents randomly from the given population and performs crossover between the parents to form a new offspring. Then new offspring possess better features from both the parents.

Let P_i and Q_j be two randomly selected parents, crossover cost associated with parent P_i is C_i and cost associated with parent Q_j is C_j . Then the Total cost of n crossover is [15] given by:

$$\sum_{i=1}^{n} P_i C_i \times \sum_{j=1}^{n} Q_j C_j = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i Q_j (C_i \times C_j) \quad (10)$$

Mutation mutates certain features of randomly selected individual. Therefore mutation cost for n iterations is:

$$\sum_{i=1}^{n} P_i C_i \, .$$

Best N selection calculates the weight or rank of 2n rows and selects n rows with highest rank. Hence total cost of Best N selection is [15]:

$$W = \sum \left(\frac{f(i)P_i}{\sum f(j)P_j} \right) C_i = \sum \left(\frac{f(i)P_i}{a_j} \right) C_i \quad (11)$$

Thus total running time of Genetic algorithm is:

$$o(1) + o(n) + \sum_{i=1}^{n} \sum_{j=1}^{n} P_i Q_j \left(C_i \times C_j \right) + \sum_{i=1}^{n} P_i C_i + \sum \left(\frac{f(i)P_i}{a_j} \right) C_i$$

2) Ana;ysis of Layered Detection (12)

In Best case the detection occurs at first layer. Then there is no need of further layers to analysis the packets. So best case complexity of detection is: $o(1)$.

And in worst case detection occurs only at the last layer, so the worst case running time of detection is: $o(n)$, where n is the number of layers.

VIII. COMPARISON OF RESULTS

A. *Advantages*

1) Less False Positive Rate

Use of genetic algorithm leads to less false positive rate. Since these algorithms frequently succeed in generating solutions of high fitness, threshold value generated for each layer will be more accurate. So that, no legitimate user will be tagged as an attacker. Thereby enhancing less false positive rate.

But in case of DPR detection, it is not applicable, because here detection is based on the comparative measure of DPR value. User with less DPR value is tagged as an attacker.

2) Automatic Detection of the Attacker

Our proposed work leads to automated detection of the attacker. If the number of packets from an IP increases above the genetic threshold value for the first time , the IP will be tagged to gray list, at this stage we can't determine whether that is attempts from the attacker or continuous request from the legitimate user.

If the number of packets from the same IP increases above threshold value then that IP will be tagged to the black list.

But in the case of DPR detection, it provides only a probabilistic detection of the attacker. It considers repeatability of the connection, and IP with less DPR will be tagged as an attacker.

3) DDoS, HTTP error, Urgent Pointer, Reset Flag, U2R, R2L attacks Detection

Automated Layered approach allows the detection of Distributed Denial of service attacks, , HTTP error, Urgent pointer overflow attack, Reset Flag attack, U2R, R2L attacks etc. Each layer is responsible for detecting corresponding attacks.

But DPR detection allows only Distributed Denial of service attacks since its main criteria in degree of periodic Repeatability.

4) Less Time Complex and Less Computational Intensive

Layered approach takes only less amount of time for detection.. For eg. , when a packet reaches a system it passes through the series of layers one after the other. Once a layer detects an attack, it is added to the Black/Gray list. Then there is no need for further layers to analysis that packet. Hence it reduces the time and computational complexity.

But in case of DPR detection, DPR of each connection is calculated and then compare the DPR with other users to find the probabilistic attacker. It incurs more time and computation.

5) Simplicity

Genetic algorithms and Layered approaches are simple to implement. So the implementation of the proposed work remains simple.

But DPR detection involves many mathematical calculations for DPR calculation and so that it will be more time consuming and error prone.

IX. Conclusion and Future Work

An Genetic algorithm based layered system to detect and filter http botnet attack and to provide less false positive rate has been studied. In DPR method [3], the detection is based on the degree of periodic repeatability which results in probabilistic detection of the attacker. The Layered detection promises automated detection biased to high false positive rate [2].

Conversely, genetic algorithm based layered system produces efficient detection which lowers false positive rate. Genetic threshold is calculated for each layer and if the numbers of packets are more than the genetic threshold value, the corresponding attack is reported and the database is updated accordingly.

In particular, it is found that such a system would be less computational intensive and more accurate. For eg. , when a packet reaches a system it passes through the series of layers one after the other. Once a layer detects an attack, it is added to the Black/Gray list. Then there is no need for further layers to analysis that packet. Layered approach provides efficiency and reliability and the automation process of grey list and black list of the firewall provides robustness.

It is further opined that rather than the active termination in GA, the cooperation of learned termination and enhanced convergence can lead to more optimized results.

Acknowledgement

The authors wish to thank the CSE department for their support and help in completing this work.

References

[1] Seena Elizebeth Mathew, Abdul Ali, “Automated Layered HTTP botnet Defence Mechanism”. International Journal of Scientific and Engineering Research,August 2013.

[2] B. Abdullah, I. Abd-alghafar, Gouda I. Salama, A. Abd-alhafez,” Performance Evaluation of a Genetic Algorithm Based Approach to Network Intrusion Detection System” International Conference on aerospace sciences & aviation technology,May 2009.

[3] Tung-Ming Koo, Hung-Chang Chang,Guo-Quan Wei, “Construction p2p firewall HTTP-Botnet Defence mechanis,”2011.

[4] Jeanne Meserve, “Official: International Hackers Going After U.S. Networks”, CNN.com, Oct 19,2007

[5] Hyunsang Choi, Hanwoo Lee, Heeko Lee, Hyorgon Kim, “Botnet Detection by Monitoring Group Activities in DNS Traffic”, 7th IEEE ICCIT, 2007, pp. 7 15-720.

[6S.N.Sivanandam, S.N.Deepa. “Introduction to Genetic Algorithms”. Springer-Verlag Berlin Heidelberg 2008

[7] Bonepalli uppalaiah, Nadipally Vamsi Krishna, Renigunta Rajendher, “Layer Based Intrusion Detection Sysytem for Network Security”,Nov 2011.

[8] L. Miller and David E. Goldberg, “Genetic Algorithms, Tournament Selection, and the Effects of Noise”1995.

[9] Blickle, Loather Thiele.” A Comparison of Selection Schemes used in Genetic Algorithm”, Swiss Federal Institute of Technology (ETH) Gloriastresse 35, 8092 Zurich,Nr. 11,December 1995.

[10] http://en.wikipedia.org/wiki/Hollands_schema_theorem.

[11] Melanie Mitchell,” An Introduction to Genetic Algorithms. 1996.”

[12] Yuri Rabinovich, Avi Wigderson,”An Analysis of a Simple Genetic Algorithms” , 1991.

[13] Barry N. Taylor and Chris E. Kuyatt, “Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results”, 1994.

[14] Gerard E. Dallal, “Degree of Feedom”, May 2007. http://www.tufts.edu/-gdallaIldof.htm

[15] NIST/SEMATECH, e-Handbook of Statistical Methods, June 2003.
[16] Don Marti, “[linux-elitists] ICANN frenzy!”, March,2001.
[17] Internet Systems Consortium, “Berkeley Internet Name Domain (BIND)”.
[18] http://www.elsevierdirect.com/companions/DNS.pdf

BIBILOGRAPHY

Miss. Seena Elizebeth Mathew completed her B.Tech from Baselios Mathews II college of Engineering, India in 2011 under Kerala University. She did her Post Graduation (M. Tech) in Computer Science and Engineering from Ilahia College of Engineering and Technology under the Mahatma Gandhi University, Kerala, India. Her areas of interests are network security and algorithms.

Mr. Abdul Ali is an Assistant Professor in the Computer Science and Engineering Department of Ilahia College of Engineering and Technology, Kerala, India. He did B.Tech in 2007 from Mahatma Gandhi University College of Engineering, Kottayam, Kerala, India followed by his M.Tech Post Graduation at Center for Information Technology and Engineering,under M S University, Tamilmadu, India in 2010. His research areas are Digital Image Processing and Modern computer Security.

Professor Dr.Janahanlal Stephen is the research Dean in the Computer Science and Engineering Department of Ilahia College of Engineering and Technology, Kerala, India. He took his Ph.D from Indian Institute of Technology (IIT), Chennai, India. His research interests are in the area of system dynamic simulation by Prof. J.W.Forrester (formely of MIT, USA), cloud computing, image processing and security.

© 2014 ACEEE
DOI: 01.IJNS.5.1.4